Claims

1. A method (500) for communication between a client computer (900) and a server computer (901), both computers (900, 901) using the hypertext transfer protocol (HTTP), the client computer (900) using an HTTP-browser (210);

the method (500) comprising the following steps:

sending (520) a first request (230) from the client computer (900) to the server computer (901);

upon receiving (530) the first request (230), the server computer (901), (i) allocating (531) a resource (340) at the server computer (901), the resource (340) with an identifier (350), and (ii) returning (532) a predetermined close instruction (360) to the browser (210), the close instruction (360) carrying the identifier (350);

upon unloading (540) the close instruction (360) from the browser (210) of the client computer (900), sending (560) a second request (240) from the client computer (900) to the server computer (901), the second request (240) carrying the identifier (350) and indicating to de-allocate the resource (340); and

upon receiving (570) the second request (240) from the client computer (900), by the server computer (901) de-allocating (580) the resource (340).

2. The method (500) of claim 1, wherein after the server computer (901) has returned (532) the predetermined close instruction (360), and before the server computer (901) receives (570) the second request (240) from the client computer (900), the server computer (901) consecutively sends content pages (335) to the client computer (900).

3. The method (500) of claim 2, wherein in the step returning (532) a predetermined close instruction (360), the browser (210) presents the close instruction (360) in a first frame (215) and presents the content pages (335) in a second frame (216).

4. The method (500) of claim 2, wherein the close instruction (360) prevents selected content pages (335) from being cached by the browser (210).

5. The method (500) of claim 1, wherein in the step sending (560) a second request (240), the client computer (900) sends the second request (240) to a predetermined address of the server computer (901).

6. The method (500) of claim 1, wherein in the step returning (532) a predetermined close instruction, the predetermined close instruction (360) comprises script (1-5).

7. The method (500) of claim 1, wherein in the step returning (532) a predetermined close instruction, the script does not lead to a presentation by the browser (210).

8. A computer program product (100/101) for HTTP-communication between a client computer (900) and a server computer (901), wherein the client computer (900) has a browser (210), the computer program product (100/101) having program code portions that cause a client processor (910) in the client computer (900) and a server processor (911) in the server computer (901) to control the communication, the computer program product (100/101) comprising:

code portions that cause the client processor (910) to send (520) a first request (230) to the server computer (901);

code portions that - upon receiving (530) the first request (230) by the server computer (901) - cause the server processor (911) to (i) allocate (531) a resource (340) at the server computer (901), the resource (340) with an identifier (350), and to (ii) return (532) a predetermined close instruction (360) to the browser (210), the close instruction (360) carrying the identifier (350);

code portions that - upon unloading (540) the close instruction (360) from the browser (210) of the client computer (900) - cause the client processor (910) to send (560) a second request (240) to the server computer (901), the second request (240) carrying the identifier (350) and indicating to de-allocate the resource (340); and

code portions that - upon receiving (570) the second request (240) from the client computer (900) - cause the server processor (911) to de-allocate (580) the resource (340).

9. The computer program product (100/101) of claim 8, wherein the code portions cause the client processor (900) to provide such a close instruction (360) that the browser (210) provides a first frame (215) to present the close instruction (360) in a first frame and provides a second frame (216) to present content pages (335) that the client computer (900) receives from the server computer (900).

10. The computer program product (100/101) of claim 8, wherein the code portions cause the client processor (900) to provide such a close instruction (360) that caching selected content pages (335) by the browser (210) is prevented.

11. The computer program product (100/101) of claim 8, wherein the code portions cause the client processor (900) to provide such a close instruction (360) that the client computer (900) sends the second request (240) to a predetermined address of the server computer (901).

12. A computer readable medium (970) storing the program code portions of the computer program product (100) of claim 8 that cause the client processor (910) to operate.

13. A computer readable medium (971) storing the program code portions of the computer program product (101) of claim 8 that cause the server processor (911) to operate.

14. A computer system (999) in that a client computer (900) and a server computer (901) use HTTP for communication and in that the client computer (900) uses an HTTP-browser (210); the computer system (999) characterized in that:

    the client computer (900) sends (520) a first request (230) to the server computer (901);

    the server computer (901), upon receiving (530) the first request (230), (i) allocates (531) a resource (340), the resource (340) having an identifier (350), and (ii) returns (532) a predetermined close instruction (360) to the browser (210) of the client computer (900), the close instruction (360) carrying the identifier (350);

    the client computer (900), upon unloading (540) the close instruction (360) from the browser (210), sends (560) a second request (240) to the server computer (901), the second request (240) carrying the identifier (350) and indicating to de-allocate the resource (340); and

    the server computer (901), upon receiving (570) the second request (240) from the client computer (900), de-allocates (580) the resource (340).

15. The computer system (999) of claim 14, wherein the client computer (900) presents the close instruction (360) in a first frame (215) and presents the content pages (335) in a second frame (216).

16. The computer system (999) of claim 14, wherein the server computer (901) provides the close instruction (360) such that in the client computer (900) the close instruction (360) prevents selected content pages (335) from being cached by the browser (210).

17. A method (600) for communication between a client computer (900) and a server computer (901), both computers (900, 901) using the hypertext transfer protocol (HTTP), the client computer (900) using an HTTP-browser (210);

the method (600) comprising the following steps:

sending (601) a request (230) from the client computer (900) to the server computer (901);

upon receiving (611) the request (230), the server computer (901):

- allocating (612) a resource at the server computer (901), the resource with an identifier (350) and a time-out period (T),
- returning (613) a close instruction (360) to the client computer (900), the close instruction (360) with the time-out period (T) and the identifier (350),
- measuring (614) the time (t) during that communication between the client computer (900) and the server computer (901) is idle, and
- de-allocating (615) the resource (340) when the measured time (t) reaches the time-out period (T); and

upon receiving (602) the close instruction (360), the client computer (900)

- measuring (603) the time (t) during that the communication between the client computer (900) and the server computer (901) is idle,
- displaying (604) a warning to the user if the measured time (t) reaches a predetermined fraction (T/X) of the time-out period (T).

18. A computer program product (100/101) for controlling HTTP-communication between a client computer (900) and a server computer (901), wherein the client computer (900) has a browser (210), the computer program product (100/101) with a client program portion (100) to control a client processor (910) and a server program portion (101) to control a server processor (911), characterized in that

the client program product portion (100) causes the client processor (910) to send (601) a request (230) from the client computer (900) to the server computer (901);

upon receiving (611) the request (230) by the server computer (901), the server program portion (101) causes the server processor (911) to allocate (612) a resource at the server computer (901), the resource with an identifier (350) and a time-out period (T), to return (613) a close instruction (360) to the client computer (900), the close instruction (360) with the time-out period (T) and the identifier (350), to measure (614) the time (t) during that communication between the client computer (900) and the server computer (901) is idle, and to de-allocate (615) the resource (340) when the measured time (t) reaches the time-out period (T); and upon receiving (602) the close instruction (360) by the client computer (900), the client program portion (100) causes the client processor (910) to measure (603) the time (t) during that the communication between the client computer (900) and the server computer (901) is idle, and to display (604) a warning to the user if the measured time (t) reaches a predetermined fraction (T/X) of the time-out period (T).

19. A method (700) for communication between a client computer (900) and a server computer (901), both computers (900, 901) using the hypertext transfer protocol (HTTP), the client computer (900) using an HTTP-browser (210);

the method (700) comprising the following steps:

sending (720) a first request (230) from the client computer (900) to the server computer (901);

allocating (731) a resource (340) at the server computer (901), the resource (340) with an identifier (350);

returning (732) a predetermined response page to the browser (210), the response page carrying the identifier (350) and carrying browser instructions;

as instructed by the response page, periodically sending (760) the second requests (240) by the browser (210) to the server computer (901), the second requests (240) carrying the identifier (350); and

at the server computer (901), periodically checking (770) the arrival of the second requests (240) with the identifier (350) from the client computer (900) and de-allocating (780) the resource (340) in case a predetermined time period (T) has lapsed since the last arrival.